

Статистическое моделирование ядра вероятностного распределения и его применение к решению задачи квантильной оптимизации с билинейной функцией потерь

Ардабьевский П.А.*

Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: <https://orcid.org/0000-0002-6857-7485>
e-mail: ardpavel@gmail.com

Гончар Д.А.**

Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: <https://orcid.org/0000-0002-2414-491X>
e-mail: dima.gonchar.29.08.13@gmail.com

Кан Ю.С.***

Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: <https://orcid.org/0000-0001-9950-4716>
e-mail: yu_kan@mail.ru

Для цитаты:

Ардабьевский П.А., Гончар Д.А., Кан Ю.С. Статистическое моделирование ядра вероятностного распределения и его применение к решению задачи квантильной оптимизации с билинейной функцией потерь // Моделирование и анализ данных. 2020. Том 10. № 3. С. 69–84. DOI: <https://doi.org/10.17759/mda.2020100306>

**Ардабьевский Павел Александрович*, магистрант, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0002-6857-7485>, e-mail: ardpavel@gmail.com

***Гончар Дмитрий Александрович*, магистрант, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0002-2414-491X>, e-mail: dima.gonchar.29.08.13@gmail.com

****Кан Юрий Сергеевич*, д.ф.-м.н., профессор, Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0001-9950-4716>, e-mail: yu_kan@mail.ru



В статье рассматривается плоская задача квантильной оптимизации с билинейной функцией потерь, которая с использованием достаточных условий оптимальности сводится к задаче линейного программирования. Сведение основано на использовании полиэдральной модели ядра вероятностного распределения вектора случайных параметров. Для построения этой модели предлагается алгоритм, основанный на методе статистического моделирования. Приводится описание программного комплекса построения модели ядра для ряда вероятностных распределений случайных параметров.

Ключевые слова: квантильная оптимизация, функция потерь, стохастическое программирование, ядро вероятностного распределения, минимаксная задача.

ВВЕДЕНИЕ

В настоящее время задачи квантильной оптимизации относятся к интенсивно развивающейся теории стохастического программирования с вероятностными критериями [1]. Функция квантили определяется как квантиль заданного уровня для вероятностного распределения некоторой функции потерь, зависящей от вектора оптимизируемых переменных и вектора случайных параметров. Оптимизационные модели с квантильным критерием используются в приложениях для учета риска или требований надежности при наличии неопределённостей стохастической природы. В теории стохастического программирования задачи с билинейными функциями потерь занимают особое место, поскольку порождены экономической проблематикой – задачами оптимизации портфельных инвестиций с учетом риска [2–4].

Явные аналитические выражения для функции вероятности в содержательных практических задачах, как правило, получить не удастся. Более того, часто не удастся даже вычислить значения вероятностных критериев с любой наперед заданной точностью. В силу указанных причин при разработке численных методов оптимизации для этих критериев приходится использовать различные их аппроксимации. Например, аппроксимация может строиться на основе построения детерминированных границ для рассматриваемых функций. Суть предлагаемого в статье метода в том, что при выполнении некоторых достаточных условий оптимальности [5] задача квантильной оптимизации оказывается эквивалентной минимаксной, где максимум берется от функции потерь на ядре вероятностного распределения вектора случайных параметров, а минимум – по оптимизируемой стратегии. Ядро определяется как пересечение всех выпуклых и замкнутых доверительных множеств в пространстве реализаций этого вектора. Этим используемый в статье подход отличается от подхода работы [4], где вместо достаточных условий оптимальности предлагаются более жесткие и трудно проверяемые требования регулярности ядра. В статье предлагается способ построения сколь угодно точных статистических аппроксимаций ядра с помощью выпуклых многогранников. В этом заключается отличие предлагаемого способа аппроксимации ядра от алгоритмов, реализованных в [6], где предлагаются детерминированные полиэдральные аппроксимации. Использование этих статистических ап-



проксимаций вместо ядра в указанной выше минимаксной задаче, как и в [4], приводит к задаче линейного программирования с большим числом ограничений.

Целью настоящего исследования является отработка альтернативных алгоритмов моделирования ядра с целью развития результатов работы [4], а также разработка программного комплекса, реализующего эти алгоритмы, позволяющего исследовать их свойства на различных примерах академического характера.

ПОСТАНОВКА ЗАДАЧИ

Введем ряд обозначений. Пусть η – случайная величина с функцией распределения $F(x) = \tilde{N}(\eta \leq x)$. Квантилью заданного уровня $\alpha \in (0, 1)$ называется величина $[\eta]_\alpha = \min\{x : F(x) \geq \alpha\}$.

Рассматривается задача квантильной оптимизации:

$$\varphi_\alpha(u) = [u_0 b + u_1 \xi_1 + u_2 \xi_2]_\alpha \rightarrow \min_{\substack{u_0 \geq 0, u_1 \geq 0, u_2 \geq 0 \\ u_0 + u_1 + u_2 = 1}}$$

где $u = (u_0, u_1, u_2)$ – вектор оптимизируемых переменных, называемый ниже стратегией, $\xi = (\xi_1, \xi_2)$ – двумерный случайный вектор, компоненты которого независимы и одинаково распределены в соответствии с заданным законом распределения, $b \geq 0$ – детерминированная скалярная величина.

Данная задача может быть проинтерпретирована как экономическая задача формирования портфеля ценных бумаг с учетом риска. В этих терминах стратегия u определяет структуру инвестиционного портфеля, компоненты вектора ξ – доходности двух рисковых ценных бумаг, b – безрисковую ставку процента. При моделировании риска с помощью квантильного критерия считается [2,3], что он ограничивается допустимой величиной $1 - \alpha$.

ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ РЕЗУЛЬТАТЫ

Алгоритмы решения поставленной задачи основаны на использовании понятия α -ядра распределения случайного вектора. Для случайного вектора ξ это ядро определяется соотношением

$$K_\alpha = \bigcap_{\|c\|=1} \left\{ x : c^T x \leq \left[c^T \xi \right]_\alpha \right\}.$$

Введем в рассмотрение функцию максимума

$$\psi_\alpha(u) = \max_{x \in K_\alpha} f(u, x),$$

где $f(u, x) = u_0 b + u_1 x_1 + u_2 x_2$, и вспомогательную минимаксную задачу

$$\psi_\alpha^* = \psi_\alpha(u_\alpha) = \min_{u \in U} \psi_\alpha(u),$$

в которой множество U допустимых стратегий то же самое, что и в рассматриваемой задаче квантильной оптимизации. В [1] установлено, что если ядро удовлет-



воряет некоторому условию регулярности, то $\varphi_\alpha(u) = \psi_\alpha(u)$ для всех $u \in U$. Поэтому вспомогательная задача оказывается эквивалентной исходной задаче квантильной оптимизации. Практическая проверка выполнения условия регулярности в настоящее время представляется нерешенной проблемой. Исключением является случай эллиптически симметричных распределений, указанный в [1]. В [5] доказано, что достаточным условием оптимальности решения вспомогательной задачи в исходной задаче квантильной оптимизации является выполнение условия

$$\tilde{N}(f(u_\alpha, \xi) \leq \psi_\alpha^*) \geq \alpha,$$

которое можно проверять, например методом Монте-Карло после решения вспомогательной задачи. При этом условие регулярности может быть не выполненным. Разработанный в рамках настоящего исследования и описанный ниже программный комплекс имеет своей целью провести апробацию такого подхода для различных законов распределения компонент случайного вектора ξ .

Приведем вспомогательную задачу к задаче линейного программирования (ЗЛП). Так как вектор u и константа b являются детерминированными, то

$$[u_0 b + u_1 \xi_1 + u_2 \xi_2]_\alpha = u_0 b + [u_1 \xi_1 + u_2 \xi_2]_\alpha.$$

Допустим, что α -ядро удалось аппроксимировать выпуклым полиэдром K_α^N . Впервые алгоритмы построения таких аппроксимаций предложены в [4] и реализованы в двумерном случае для ряда распределений в [6]. При замене -ядра его аппроксимацией во вспомогательной задаче получаем следующую минимаксную задачу

$$u_0 b + \max_{x \in K_\alpha^N} (u_1 x_1 + u_2 x_2) \rightarrow \min_{\substack{u_0 \geq 0, u_1 \geq 0, u_2 \geq 0 \\ u_0 + u_1 + u_2 = 1}}$$

которая согласно [4] эквивалентна ЗЛП вида

$$y \rightarrow \min_{\substack{u_0 \geq 0, u_1 \geq 0, u_2 \geq 0 \\ u_0 + u_1 + u_2 = 1 \\ u_0 b + u_1 x_{j1} + u_2 x_{j2} \leq y, j = \overline{1, N}}}$$

в которой $x_j = (x_{j1}, x_{j2})$, $j = \overline{1, N}$ – все вершины полиэдра K_α^N .

СТАТИСТИЧЕСКИЙ АЛГОРИТМ АППРОКСИМАЦИИ ЯДРА

Будем строить статистическую оценку полиэдральной аппроксимации -ядра в виде

$$K_\alpha^N = \bigcap_{i=1}^N \{x : c_i^T x \leq \widehat{[c_i^T \xi]_\alpha}\},$$

где «крышка» обозначает выборочную оценку квантили [1]. Для ее построения необходимо сначала сгенерировать векторы единичной нормали $c_i, i = \overline{1, N}$, где N – число ограничений. В [1] было предложено генерировать векторы c_i с помощью нанесения равномерной сетки на поверхность единичной окружности. Векторы c_i равны $(\cos \varphi_i, \sin \varphi_i)^T$, где φ_i – угол -той части разбиения окружности.



В отличие от работ [4,6] предлагается использовать случайную величину φ , равномерно распределенную по закону $R(0, 2\pi)$. Сгенерировав выборку угла φ объемом N , построим вариационный ряд полученных реализаций, а затем вычислим $\cos\varphi_i$ и $\sin\varphi_i$, где φ_i – i -ый член построенного вариационного ряда полученной выборки реализаций случайной величины φ .

Получив набор векторов единичной нормали c_i , построим выборочные оценки правых частей ограничений. Для этого генерируем выборку вектора ξ объемом L и для каждого вектора c_i построим вариационный ряд выборки (c_i, ξ_j) , $j = 1, L$. Выборочной оценкой для каждого линейного ограничения будет служить αL -ый (функция округления дробного числа до ближайшего целого вправо) член соответствующего вариационного ряда. Таким образом получена система линейных ограничений.

Этот способ основывается на том, что при генерации векторов c_i был построен вариационный ряд случайной величины φ , потому что в этом случае мы получаем систему прямых, где каждая следующая прямая является «соседом» предыдущей, то есть если эти прямые являются сторонами аппроксимирующего многогранника, то их точка пересечения будет вершиной этого многогранника или, если по-другому, они являются соседними сторонами. К сожалению, из-за того, что правые части линейных ограничений вычисляются не точно, а в соответствии с выборочными (статистическими) оценками, то некоторые ограничения могут оказаться лишними. В результате число вершин аппроксимирующего полиэдра может оказаться меньше N , что отличает предлагаемый алгоритм от алгоритмов [4,6] и существенно затрудняет исследование его сходимости.

Итак, имеется множество линейных ограничений $c_i^T x \leq \widehat{[c_i^T \xi]}_\alpha$, $i = \overline{1, N}$, полученных в результате алгоритма генерации линейных ограничений. Прямые $c_i^T x = \widehat{[c_i^T \xi]}_\alpha$, $i = \overline{1, N}$, которые задают эти линейные ограничения, являются сторонами аппроксимирующего многогранника, причем они расположены в таком порядке, что каждая следующая прямая $c_{i+1}^T x = \widehat{[c_{i+1}^T \xi]}_\alpha$ является «соседом» предыдущей. Это обстоятельство позволяет серьезно снизить количество решаемых систем линейных уравнений по сравнению с алгоритмом, используемым в [4].

Шаг 1. Составляем и решаем системы линейных уравнений $c_i^T x = \widehat{[c_i^T \xi]}_\alpha$ (для первой итерации алгоритма $i = 1$) и прямых $c_j^T x = \widehat{[c_j^T \xi]}_\alpha$, $j = i + 1, N$. Решаем до тех пор, пока не будет найдена точка пересечения, удовлетворяющая всем линейным ограничениям, или пока не переберем все возможные системы. Если подходящая точка была найдена, то прямые $c_k^T x$ и $c_k^T x = \widehat{[c_k^T \xi]}_\alpha$, где k – номер первой встретившейся прямой, точка пересечения с которой удовлетворяет всем линейным ограничениям, являются сторонами многогранника, их точка пересечения является его вершиной, а все прямые $c_i^T x = \widehat{[c_i^T \xi]}_\alpha$, $i = \overline{2, k-1}$, не являются сторонами аппроксимирующего многогранника, переходим к шагу 1, $i = k$. Если подходящих точек найдено не было, и прямая ранее не была отмечена как сторона аппроксимирующего многогранника, переходим к шагу 2. Если же для нее уже есть одна подходящая точка, переходим к шагу 3.



Шаг 2. Если прямая $c_i^T x = \left[c_i^T \xi \right]_\alpha$ не была ранее отмечена как сторона многогранника, то эта прямая ей не является. Переходим к шагу 1, $i = i + 1$.

Шаг 3. Если ранее для прямой $c_i^T x = \left[c_i^T \xi \right]_\alpha$ была найдена одна вершина, то есть эта прямая является стороной аппроксимирующего многогранника, пересекаем эту прямую с самой первой подошедшей прямой. Найденная точка будет являться вершиной искомого многогранника. Конец алгоритма.

Как видно в этом алгоритме, мы не решаем все C_n^2 линейных систем уравнений, как это было в [4] даже в самом плохом случае. При высоком объеме выборки L случайного вектора ξ ниже на примерах увидим, что сторонами аппроксимирующего многогранника являются практически все или все прямые, задающие линейные ограничения. В таком случае количество решаемых линейных систем становится равно N .

ОПИСАНИЕ ПРОГРАММНОГО КОМПЛЕКСА

Рассмотрим интерфейсное окно:

Количество векторов

Количество ограничений

Уровень надежности на отрезке [0; 1]

Детерминированная константа

Логнормальное распределение
 Нормальное распределение
 Равномерное распределение
 Кусочное равномерное
 Распределение Коши
 Экспоненциальное распределение
 Гамма распределение
 Распределение Рэлея
 Распределение Вейбулла

Первый параметр Второй параметр

Для расчета объема выборки

Бетта Эпсилон

Рис. 3.1. Окно ввода данных для работы программы

В интерфейсе требуется указать следующее:

- Количество векторов – количество сгенерированных случайным образом векторов, из которых будет строиться ядро
- Количество ограничений – количество случайным образом сгенерированных векторов распределения $R(0, 2\pi)$
- Уровень надежности – квантиль полученной выборки векторов
- Детерминированная константа – постоянная величина, используемая при решении ЗЛП
- Выбор одного из распределений
- Ввод параметров распределения
- Бетта и эпсилон – величины, используемые для расчёта рекомендуемого объема выборки
- Пуск – кнопка запуска расчёта

Вывод программы, а также описание выводимых полей будет приведено в «4. Результаты работы программы»

Проанализируем время работы программы.

Так как пользователь будет желать получить точную аппроксимацию, то он будет вводить большое количество векторов и ограничений. Рассмотрим таблицу времени генерации векторов и ограничений при разных задаваемых параметрах:



Таблица 3.1

Генерация векторов и ограничений

Кол-во векторов	Кол-во ограничений	Время генерации (сек)
10 ⁴	100	0.28
10 ⁴	200	0.46
10 ⁴	300	0.6
10 ⁵	100	39
10 ⁵	200	67.57
10 ⁵	300	91.17
10 ⁶	100	496.17
10 ⁶	200	890.92
10 ⁶	300	1311.97

Опираясь на результаты, приведенные в таблице 3.1, можно сказать, что оба вводимых параметра значимо влияют на время работы. Далее мы скажем оптимальные параметры для получения результата. Сейчас скажем только то, что кол-во векторов должно варьироваться от 10⁴ до 10⁶, так как при меньших значениях отбрасывается слишком много прямых, а при больших время работы программы слишком велико.

Теперь проведем анализ сложности отбрасывания прямых. По данному шагу были проанализированы два варианта отброса.

В первом варианте нам потребовалось перебрать все точки пересечения прямых (сложность $O(\frac{N!(N-1)!}{2})$), для каждой точки необходимо было решить систему (сложность $O(N^2)$), а затем отсортировать полученные вершины (сложность $O(N)$).

Во втором случае мы лишь сортируем прямые (сложность $O(N)$) и находим точки проходя по всем прямым (сложность $O(N)$).

Таблица ниже демонстрирует разницу в работе:

Таблица 3.2

Нахождение вершин многогранника

Кол-во векторов	Кол-во ограничений	Время нахождения вершин (без сортировки прямых / с сортировкой прямых) (сек)
10 ⁴	100	7.34 / 2.58
10 ⁴	200	41.04 / 16.76
10 ⁴	300	234.19 / 53.45
10 ⁵	100	6.32 / 2.04
10 ⁵	200	41.81 / 17.08
10 ⁵	300	240.75 / 52.14
10 ⁶	100	7.2 / 2.78
10 ⁶	200	42.89 / 16.95
10 ⁶	300	54.96 / 18.83

Из таблицы можно заметить, что количество введенных векторов не влияет на время отбрасывания прямых.



Для краткости описания введем обозначения:

- L – количество векторов
- N – количество ограничений N

Рассмотрим теперь таблицу, в которой приведены итоги работы конечной программы:

Таблица 3.3

Результат работы

L	N	Время генерации (сек)	Время отброса прямых и нахождения вершин (сек)	Кол-во вершин многогранника
10^4	100	0.28	2.58	63
10^4	200	0.46	16.76	72
10^4	300	0.6	53.45	102
10^5	100	39	2.04	76
10^5	200	67.57	17.08	108
10^5	300	91.17	52.14	127
10^6	100	496.17	2.78	88
10^6	200	890.92	16.95	149
10^6	300	1311.97	54.96	189

Также нельзя не заметить, что время построения аппроксимации ядра всегда варьируется в пределах [5, 6] секунд

Оптимальные параметры пользователь может выбрать таким образом: если более важно время работы программы, то стоит выбрать $L=10^5$ и $N=250$; если более важна точность, то выбрать $L=10^6$ и $N=200$.

РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

В результате работы мы можем увидеть следующее: (см. рис. 4.1)

Опишем выводимые колонки.

Первая колонка – повторный ввод данных. Она необходима для изменения введенных параметров.

Вторая колонка – результаты решения ЗЛП с дополнительными данными. Опишем их:

Таблица – значения u_0 , u_1 , u_2

Значение целевой функции – значение при расчёте ЗЛП

Расчетное значение вероятности по методу Монте-Карло

Строка, в которой написано является ли решение оптимальным

Количество векторов – введенное количество векторов

Количество вершин многогранника – количество вершин ядра, получившееся в результате отброса прямых ограничениями

Следующие три строки описывают время работы программы

N , Nr – оптимальные размеры выборки

Третья колонка отображает построенное ядро

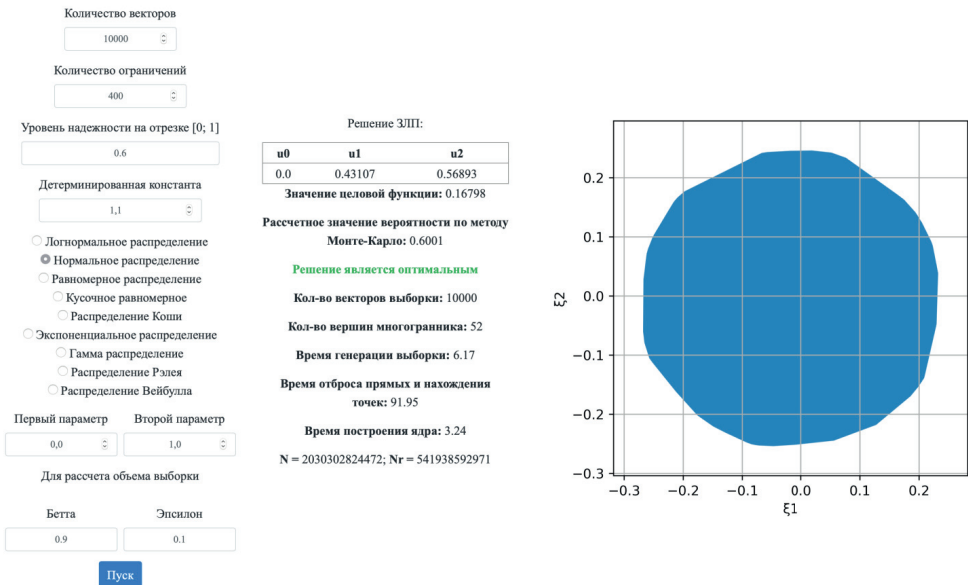


Рис. 4.1. Результат работы программного комплекса

Рассмотрим случай равномерного распределения компонент вектора ξ на отрезке $(-1, 1)$, $\alpha = 0.9$, количество ограничений $N = 50$.

На рис. 4.2 и рис. 4.3 видно, что с увеличением объема выборки L увеличивается и точность аппроксимации хоть и несильно, и результат сходится к аналитическому результату. Если бы в разработанной программе была возможность настраивать распределение каждой компоненты вектора ξ отдельно, то схожесть форм в случае равномерного распределения компонент аналитически построенного аппроксимирующего многогранника и построенного на основе выборочной оценки была бы заметна куда больше.

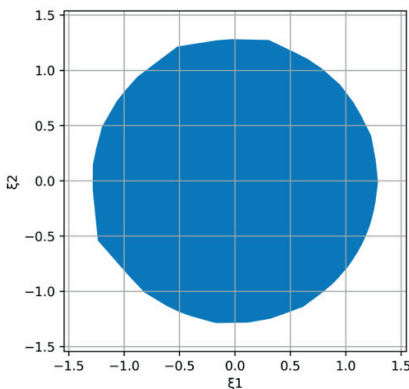


Рис. 4.2 Результат аппроксимации для равномерного распределения, $L = 10^4$

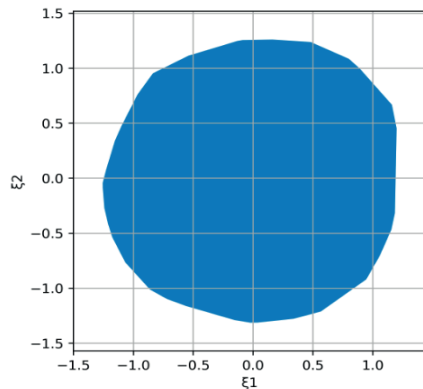


Рис. 4.3 Результат аппроксимации для равномерного распределения, $L = 10^6$



Теперь построим аппроксимации для нормального распределения $N(0,1)$ компонент вектора ξ , $\alpha = 0.9$, количество ограничений $N = 50$.

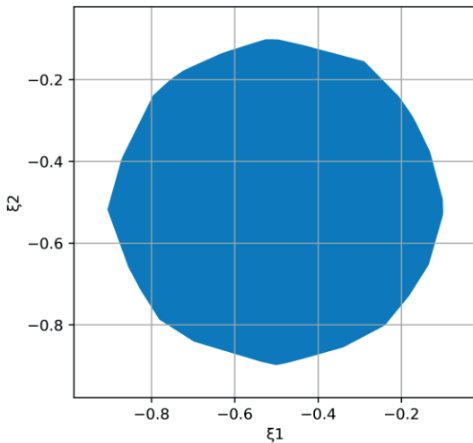


Рис. 4.4 Результат аппроксимации для нормального распределения, $L=10^4$

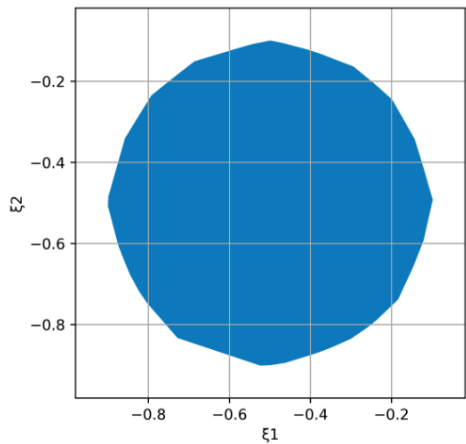


Рис. 4.5 Результат аппроксимации для нормального распределения, $L=10^6$

Как видно на рис. 4.4 и рис. 4.5 результат так же, как и в случае с равномерным распределением, все больше становится похож на полученный аналитически с увеличением объема выборки L . Даже неровность формы из-за случайной природы векторов c_i сглаживается и сходится к ровному кругу.

Результат для распределения Коши отличается по своему характеру от результатов, полученных для равномерного и нормального распределений. Было использовано стандартное распределение Коши для обеих компонент вектора ξ , $\alpha = 0.9$, $N = 100$.

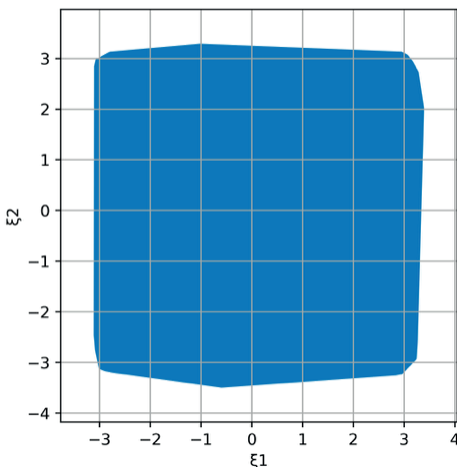


Рис. 4.6 Результат аппроксимации для распределения Коши, $N = 50$, $L=10^4$

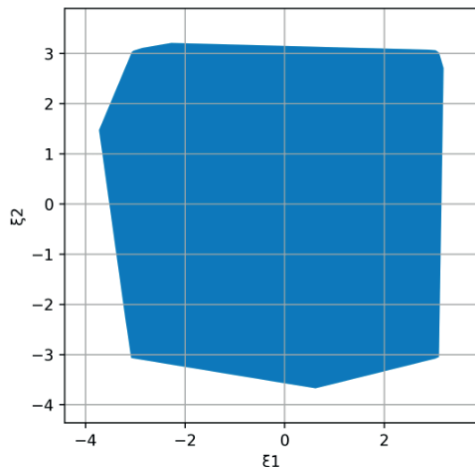


Рис. 4.7 Результат аппроксимации для распределения Коши, $N = 50$, $L=10^5$

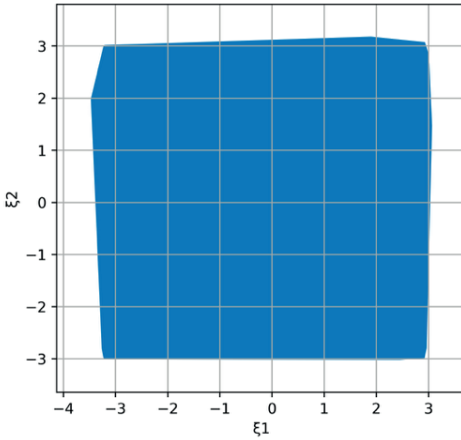


Рис. 4.8 Результат аппроксимации для распределения Коши, $N=100$, $L=10^4$

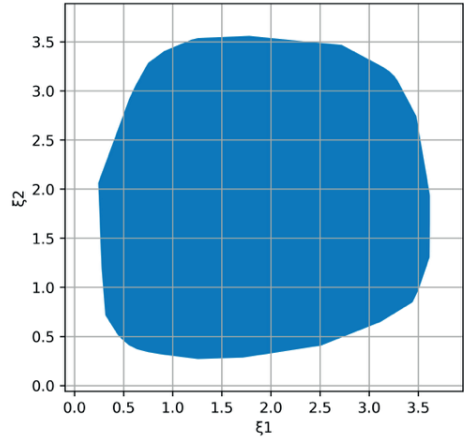


Рис. 4.9 Результат аппроксимации для логнормального распределения, $L=10^4$

Как оказалось, аппроксимация в случае распределения Коши плохо реагирует на повышение объема выборки L , этот эффект мы можем наблюдать, сравнив рис. 4.6 и рис. 4.7. Зато увеличение количества ограничений N увеличивает точность аппроксимации, и результат сходится к полученному аналитическому.

Аппроксимация при логнормальном и экспоненциальном распределении компонент вектора ξ схожа по своему характеру с аппроксимацией для равномерного и нормального распределений. Для построения многогранников были использованы $\text{Log}N(1)$ и $E(1)$, $\alpha = 0.9$, количество ограничений $N = 50$.

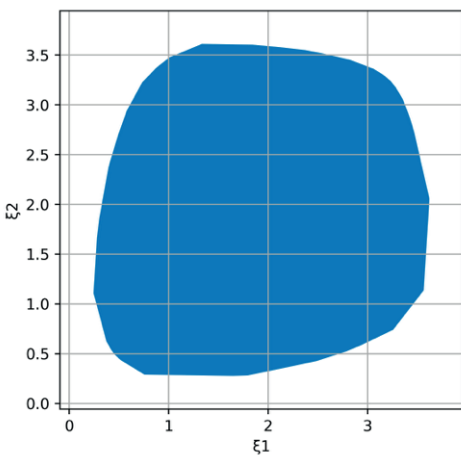


Рис. 4.10 Результат аппроксимации для логнормального распределения, $L=10^6$

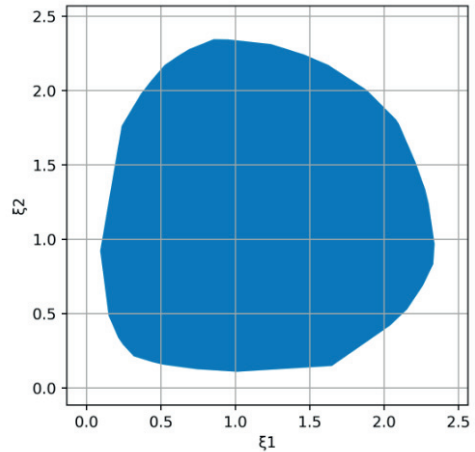


Рис. 4.11 Результат аппроксимации для экспоненциального распределения, $L=10^4$

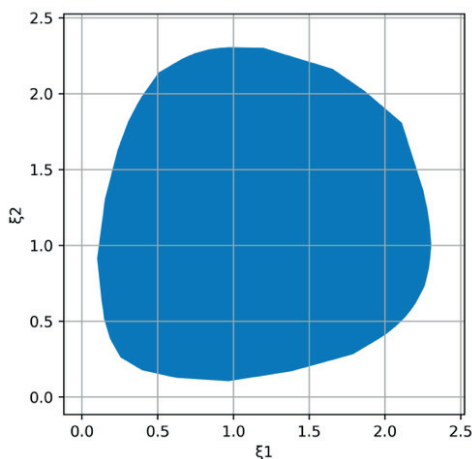


Рис. 4.12 Результат аппроксимации для экспоненциального распределения, $L=10^6$

Из полученных результатов видна правильность работы алгоритма, а также эмпирическим способом была доказана сходимость густой сетки векторов единичной нормали к истинным α -ядрам. Стоит отметить, что α не варьировалась по причине того, что она влияет только на размер α -ядра, а не на его форму (меняется только для равномерного распределения).

Ниже приведены результаты решения исходной задачи для $b=1.1$, $N=50$, (объем выборки был взят небольшим, чтобы уменьшить время работы программы), компоненты вектора ξ распределены равномерно на $(-1,1)$. Ответом на поставленную задачу является вектор стратегий u .

Решение ЗЛП:

u_0	u_1	u_2
0.08663	0.91337	0.0

Значение целевой функции y : 0.0

Кол-во векторов выборки: 10000

Кол-во вершин многогранника: 41

Время генерации выборки: 2.91

Время отброса прямых и нахождения точек: 0.38

Время построения ядра: 5.34

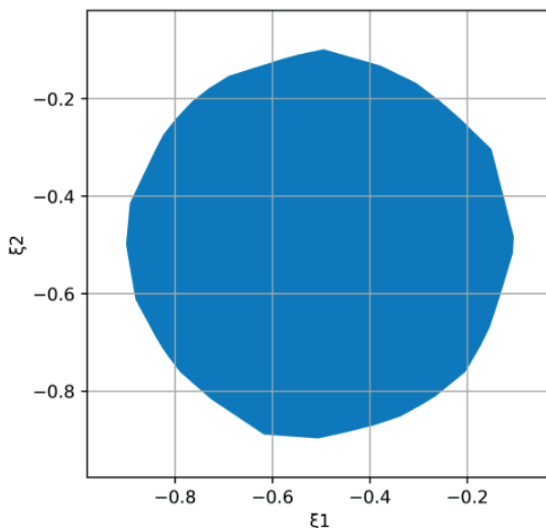


Рис. 4.13 Решение задачи при $\alpha = 0.9$



Решение ЗЛП:

u0	u1	u2
0.0	0.52357	0.47643

Значение целевой функции y : 0.76047

Кол-во векторов выборки: 10000

Кол-во вершин многогранника: 40

Время генерации выборки: 2.6

Время отброса прямых и нахождения точек: 0.32

Время построения ядра: 5.04

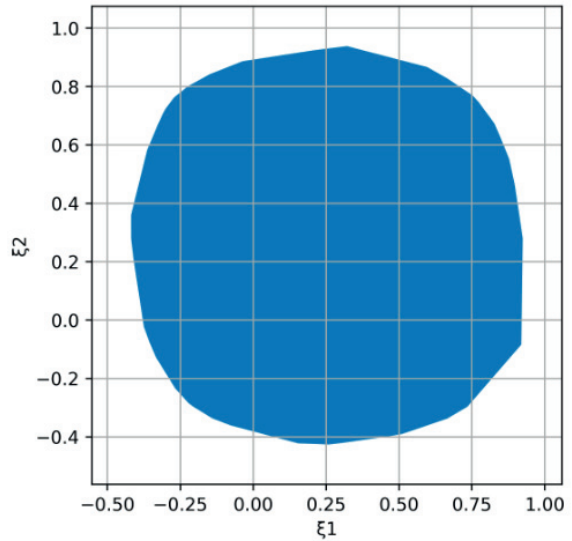


Рис. 4.14 Решение задачи при $\alpha = 0.95$

Решение ЗЛП:

u0	u1	u2
0.00178	0.99822	0.0

Значение целевой функции y : 0.0

Кол-во векторов выборки: 10000

Кол-во вершин многогранника: 32

Время генерации выборки: 2.92

Время отброса прямых и нахождения точек: 0.37

Время построения ядра: 5.23

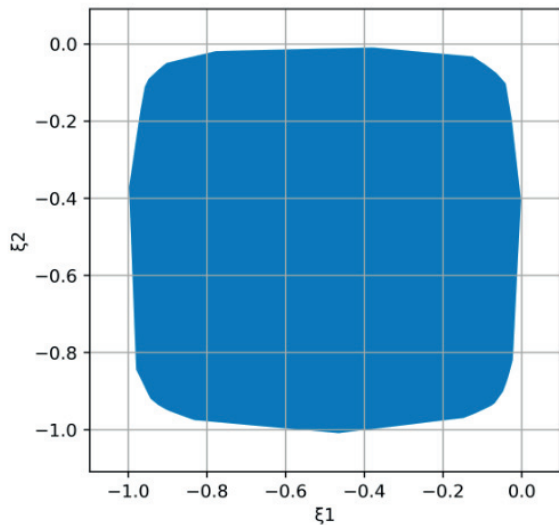


Рис. 4.15 Решение задачи при $\alpha = 0.99$

На вышеприведенных рис. 4.13, рис. 4.14 и рис. 4.15 отчетливо видно, как с ростом α форма α -ядра становится все более квадратной. Это связано с тем, что случайный



вектор ξ распределен равномерно на квадрате, и так как α -ядро с ростом α увеличивается, а квадрат, на котором распределен вектор ξ , нет, получается такая форма.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы разработана программа для решения задачи квантильной оптимизации с билинейной функцией потерь с заданным распределением случайных параметров при помощи построения аппроксимации ядра различной точности.

В программе имеются следующие изменяемые пользователем параметры:

- Количество векторов;
- Количество ограничений;
- Уровень надежности;
- Значение детерминированной константы;
- Распределение;
- Параметры распределения;
- Параметры расчёта оптимального числа векторов и ограничений.

Результаты многочисленных расчетов для различных распределений свидетельствуют о конкурентоспособности используемого статистического алгоритма аппроксимации -ядра по сравнению с алгоритмами [4,6].

Литература

1. Кибзун А.И., Кан Ю.С. Задачи стохастического программирования с вероятностными критериями. – М.: Физматлит, 2009.
2. Кан Ю.С. Формирование портфелей ценных бумаг с учетом риска. Учеб. пособие. – М.: Изд-во МАИ, 2008.
3. Кан Ю.С., Тузов Н.В. Минимизация квантили нормального распределения билинейной функции потерь. // Автоматика и телемеханика. 1998, № 11, с. 82–92.
4. Васильева С.Н., Кан Ю.С. Метод решения задачи квантильной оптимизации с билинейной функцией потерь // Автоматика и телемеханика, 2015, № 9, с. 83–101.
5. Кан Ю.С. Расширение задачи квантильной оптимизации с линейной по случайным параметрам функцией потерь // Автоматика и телемеханика, 2020, № 12 (в печати).
6. Васильева С.Н., Кан Ю.С. Алгоритм визуализации плоского ядра вероятностной меры. // Информатика и ее применения, 2018, № 2, с. 60–68.



Statistical Simulation of the Probability Distribution Kernel and Its Application to Solve the Quantile Optimization Problem with the Bilinear Loss Function

Pavel A. Ardabyevskiy*

Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: <https://orcid.org/0000-0002-6857-7485>
e-mail: ardpavel@gmail.com

Dmitry A. Gonchar**

Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: <https://orcid.org/0000-0002-2414-491X>
e-mail: dima.gonchar.29.08.13@gmail.com

Yuri S. Kan***

Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: <https://orcid.org/0000-0001-9950-4716>
e-mail: yu_kan@mail.ru

The article considers a plane quantile optimization problem with a bilinear loss function, which, using sufficient optimality conditions, is reduced to a linear programming problem. The reduction is based on the use of a polyhedral model of the kernel of the probability distribution of the vector of random parameters. To build this model, an algorithm based on the method of statistical modeling is proposed. A description of the software package for constructing a kernel model for a number of probability distributions of random parameters is given.

Keywords: quantile optimization, loss function, stochastic programming, kernel of probability distribution, minimax problem.

For citation:

Ardabyevskiy P.A., Gonchar D.A., Kan Y.S. Statistical Simulation of the Probability Distribution Kernel and Its Application to Solve the Quantile Optimization Problem with the Bilinear Loss Function. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 3, pp. 69–84. DOI: <https://doi.org/10.17759/mda.2020100306> (In Russ., abstr. in Engl.).

***Pavel A. Ardabyevskiy**, Student of the magistracy, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: <https://orcid.org/0000-0002-6857-7485>, e-mail: ardpavel@gmail.com

****Dmitry A. Gonchar**, Student of the magistracy, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: <https://orcid.org/0000-0002-2414-491X>, e-mail: dima.gonchar.29.08.13@gmail.com

*****Yuri S. Kan**, Dr.Sci.(Phys.-Math.), professor, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: <https://orcid.org/0000-0001-9950-4716>, e-mail: yu_kan@mail.ru



References

1. Kibzun A.I., Kan Y.S. Zadachi stohasticheskogo programmirovaniya s veroyatnostnymi kriteriyami. – M.: Fizmatlit, 2009.
2. Kan Y.S. Formirovanie portfelej cennyh bumag s uchetom riska. M.: MAI publishing, 2008.
3. Kan Y.S., Tuzov N.V. Minimizatsiya kvantili normal'nogo raspredeleniya bilinejnoj funkicii poter'. Avtomatika i Telemekhanika. 1998, № 11, pp. 82–92.
4. Vasil'eva S.N., Kan Y.S. Metod resheniya zadachi kvantil'noj optimizatsii s bilinejnoj funkciej poter'. Avtomatika i Telemekhanika, 2015, № 9, pp/ 83–101.
5. Kan Y.S. Rasshirenie zadachi kvantil'noj optimizatsii s linejnoj po sluchajnym parametram funkciej poter'. Avtomatika i Telemekhanika, 2020, № 12.
6. Vasil'eva S.N., Kan Y.S. Algoritm vizualizatsii ploskogo yadra veroyatnostnoj mery. Informatika i ee primeneniya, 2018, № 2, pp. 60–68.